S25: Introduction To Processor Architecture
Assignment 1

Deadline: February 1, 2025

January 25, 2025

1 Arithmetic and Logical Unit

In this section of the code, you are required to implement a 64-bit Arithmetic Logical Unit

31 25	24 20	19 15	14 12	11 7	6 0
funct7	rs2	rs1	funct3	rd	opcode
7	5	5	3	5	7
0000000	src2	src1	ADD/SLT/SLTU	dest	OP
0000000	src2	src1	AND/OR/XOR	dest	OP
0000000	src2	src1	SLL/SRL	dest	OP
0100000	src2	src1	SUB/SRA	dest	OP

Figure 1: Register-Register Integer Operations

1.1 ALU Functions

Given inputs A (src1 or rs1) ,B(src2 or rs2) and output, ADD and SUB perform addition and subtraction respectively. Overflows are ignored and the low 64 bits of results are written to the output (rd).

SLT and SLTU perform signed and unsigned compares respectively, writing 1 to output if rs1 (A) $<$ rs2 (B), 0 otherwise.

AND, OR, and XOR perform bitwise logical operations.

SLL, SRL, and SRA perform logical left, logical right, and arithmetic right shifts on the value in register rs1 by the shift amount held in the lower 5 bits of register rs2.

IMPORTANT

You are not allowed to use +, -, &, etc directly on 64 bit inputs for the 64 bit operations given. We expect you all to write each of the above modules from scratch in Verilog (structural). All input and outputs should be signed and use 2's complement for the subtraction.

Write a final wrapper ALU unit from where you will call the modules mentioned above based on the control input. The ALU unit takes as input the control signal, and two 64-bit inputs, and returns the 64-bit output corresponding to the control signal chosen.